Database Functions

Vulnerable to TOCTOU issues

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-22

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6166 bytes

<table>
<tr><td><b>Attack Category</b></td><td><ul><li>Path spoofing or confusion problem</li></ul></td></tr>
<tr><td><b>Vulnerability Category</b></td><td><ul><li>Indeterminate File/Path</li><li>TOCTOU - Time of Check, Time of Use</li></ul></td></tr>
<tr><td><b>Software Context</b></td><td><ul><li>Filename Management</li></ul></td></tr>
<tr><td><b>Location</b></td><td><ul><li>db.h</li><li>dbm.h</li></ul></td></tr>
<tr><td><b>Description</b></td><td><p>Care must be exercised when accessing files from passed in pathnames. Each of these functions takes a filename (or pathname) that can potentially be subjected to a time-of-check-to-time-of-use race condition during an access control check.</p><p>db_initialize(const char *dictionary_pathname) is used to initialize a database with a file that contains catalog information associated with that database.</p><p>dbm_open(const char *file, int open_flags, mode_t file_mode) opens the database that is specified by file.</p><p>dbminit(char *file) creates/opens the database file.db.</p><p>During the time between the check that ensures the user has access to that file and the actual access to the file, an attacker can change the filename to one of his choosing.</p></td></tr>
<tr><td><b>APIs</b></td><td><table><tr><th>FunctionName</th><th>Comments</th></tr><tr><td>db_initialize</td><td></td></tr><tr><td>dbm_open</td><td></td></tr><tr><td>dbminit</td><td></td></tr></table></td></tr>
<tr><td><b>Method of Attack</b></td><td><p>The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between</p></td></tr>
</table>

[1] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

<table>
<tr><td></td><td>the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results
An attacker could link the targeted database file to one of his choosing and gain access and control over a file he should not have privileges to.</td></tr>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
<tr><td><b>Solutions</b></td><td>
<table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>Generally applies.</td><td>The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.</td><td>Does not resolve the underlying vulnerability but limits the false sense of security given by the check.</td></tr>
<tr><td>Generally applies.</td><td>Limit the interleaving of operations on files from multiple processes.</td><td>Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.</td></tr>
<tr><td>Generally applies.</td><td>Limit the spread of time (cycles) between the check and use of a resource.</td><td>Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.</td></tr>
<tr><td>Generally applies.</td><td>Recheck the resource after</td><td>Checking the filename</td></tr>
</table>
</td></tr>
</table>

	the use call to verify that the action was taken appropriately.	permissions after the operation does not change the fact that the operation may have been exploited but it does allow halting of the application in an error state to help limit further damage.
When user specification of the file to be altered is not necessary.	Do not rely on user-specified input to determine what the file to be made executable.	This will reduce exposure but will not eliminate the problem.

Signature Details

Examples of Incorrect Code

```
#include
#include
/* check permissions to the path
*/
if(!access(file, ...){
/* open a database */
dbm_open(filename, ...)
}
else{
/* permission was denied */
}
```

Examples of Corrected Code

Source References

- Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X
- ITS4 Source Code Vulnerability Scanning Tool [2]
- http://www.phy.ohiou.edu/cgi-bin/man-cgi.sol?db_initialize+3N
- http://www.linuxvalley.it/encyclopedia/ldp/manpage/man3/db_open.3.php
- http://publib16.boulder.ibm.com/pseries/en_US/libs/commtrf1/dbminit.htm

Recommended Resource

Discriminant Set	**Operating System**	• ~~AIX IS PLACEHOLDER~~
	Language	•

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com